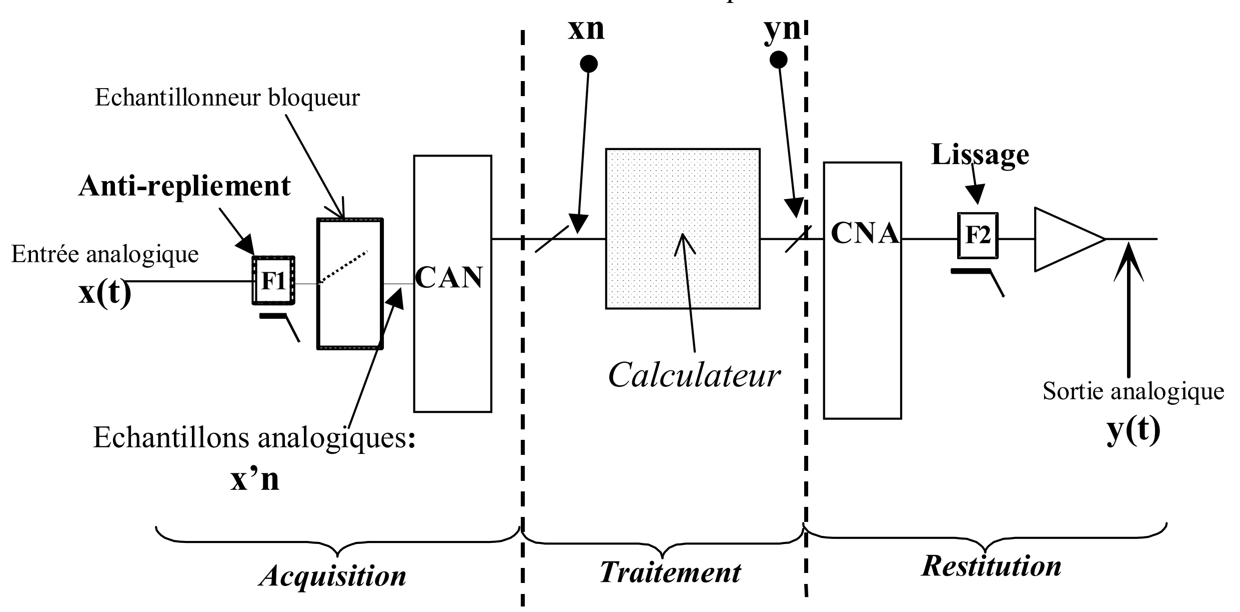
TP1 Traitement Numérique du Signal

Avec le logiciel Matlab :
Découverte de la convolution, régime transitoire
Sur DSP (processeurs de signaux) :
Échantillonnage, et reconstitution. Repliement spectre …

I Introduction : chaîne de traitement de signal en numérique

Le traitement numérique des signaux est une technique dont la théorie et les avantages sont en fait connus depuis longtemps, mais qui n'est devenue vraiment exploitable que depuis peu grâce aux progrès des composants électroniques en puissance de calcul (ordinateurs, processeurs de signaux ……). De nombreux filtres ou traitements (en particulier les filtres à phase linéaire) ne peuvent s'effectuer qu'en numérique (ou du moins qu'au moyen d'échantillons du signal).

I 1 Eléments constitutifs



L'***échantillonneur-bloqueur*** "discrètise" le signal en prélevant des échantillons à une cadence d'échantillonnage **Fech = 1/Tech** et les envoie au CAN (signal **x'n**)
Le ***CAN*** permet la **numérisation** de chaque échantillon et fournit les **xn**.
Le « **Calculateur** » (microcontroleur, ordinateur, processeur de signal, ou circuit câblé …) permet de faire des traitements sur des données numériques.
Un **signal se sortie numérique** peut être fourni sous formes d'échantillons **yn**, à la même cadence Fech de préférence.
Ces échantillons peuvent restituer un **signal de sortie analogiqu**e par passage par un **CNA** et un **filtre F2** dit de **lissage** (jouant le rôle d'interpolateur).

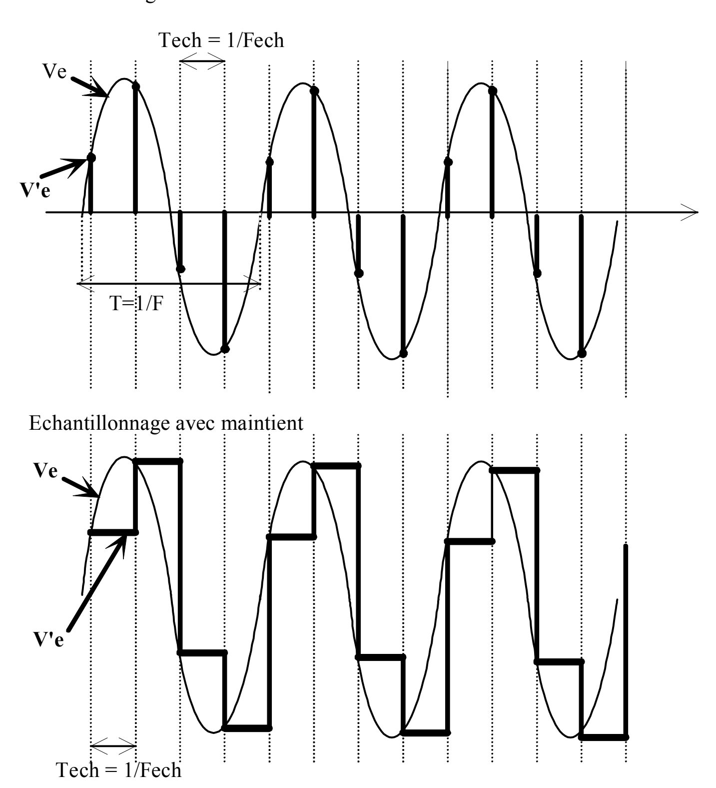
Le rôle de F1 (filtre anti-repliement) est expliqué plus loin.

I 2 Echantillonnage et reconstitution

I 2 a Rôle de l'échantillonneur bloqueur

Soit un signal x(t) sinusoïdal à échantillonner, de fréquence **F**. Et soit **Fech** la cadence de prise en compte des échantillons.

signal d'entrée sinusoidal avec F < Fech/2



Le maintient entre deux échantillons permet:

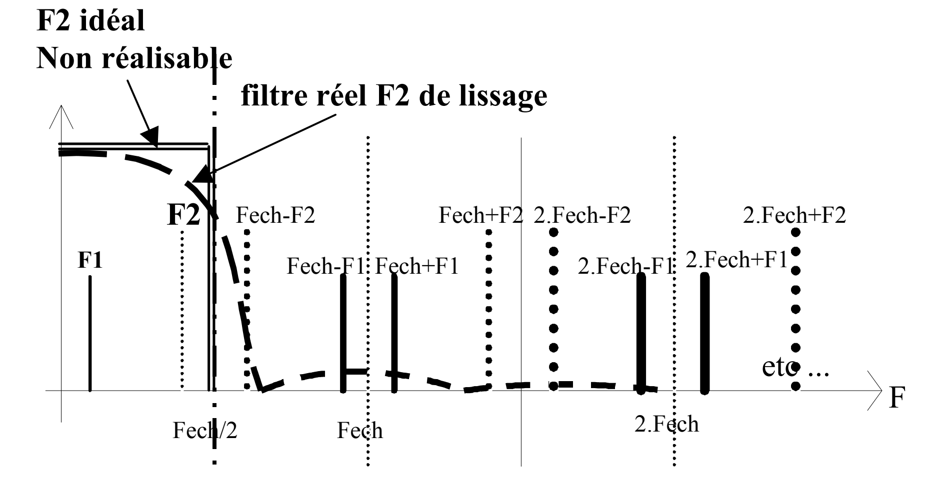
-En **acquisition** pour avoir le **temps de convertir** en binaire les valeurs

-En **restitution** pour avoir un signal **d'énergie suffisante**

I 2 b Spectre du signal échantillonné, reconstitution

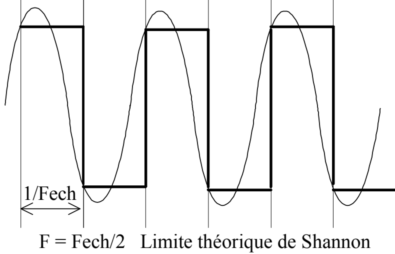
Soit un signal **Ve sinusoïdal**, de **fréquence F**, on montrerait qu'il fournit après échantillonnage toute une série de fréquences supplémentaires **Fech-F Fech+F 2.Fech-F 2Fech+F 3.Fech-F 3.Fech+F ......**

Pour deux fréquences F1 et F2 $< \frac{Fech}{2}$, on aura donc:



Tant que les raies Fech _Fi sont au delà de Fech/2, on voit donc que par suppression des fréquences élevées au delà de Fech/2, on peut reconstituer le signal de départ. Le filtre à utiliser est un **passe bas** dit de **"lissage".**

Cas limite F = Fech/2:

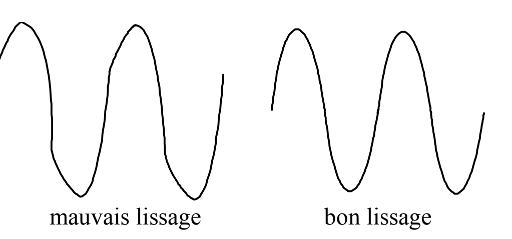


Il faut donc toujours avoir **F < Fech/2** (sauf cas très particulier de "sous échantillonnage"). C'est le **théorème de Shannon**.

➢ ***Le filtre de lissage :***

Un **filtre idéal** laisserait tout passer à Fech /2 et rien au delà, ce n'est évidemment pas possible.

La reconstitution est en fait très facile pour F1, et un peu moins pour F2 qui est plus proche de Fech/2 .



Pour **reconstituer** proprement (avec très peu de distorsions) le signal de départ jusqu'à des fréquences proches de Fech/2, le **filtre de lissage réel** devra donc être **performant :**

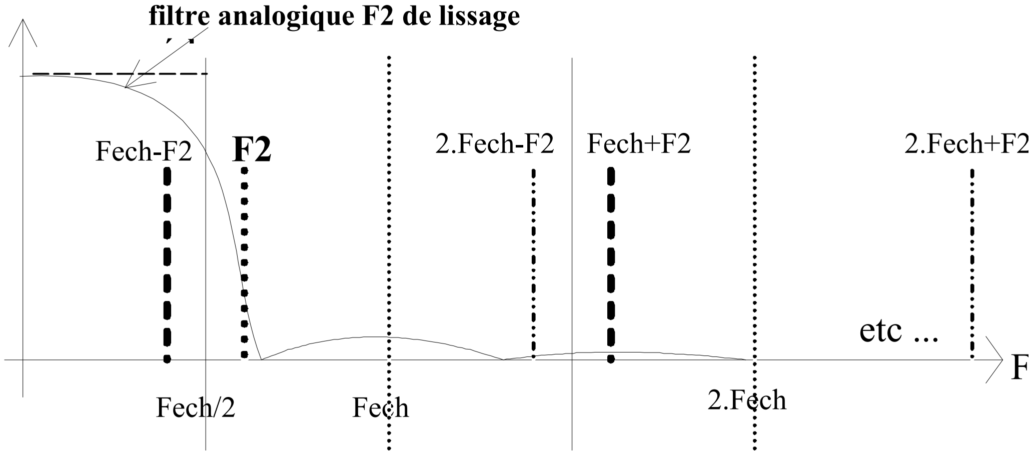
- **Le plus plat possible** ou très peu oscillant **jusqu'à presque Fech/2**
- **Atténuer déjà à Fech/2 de 40 à 50dB**.

On choisit des filtres classiques analogiques de Butterwoth ou Tchebycheff, souvent d'ordre assez élevé (6 ou 7).

I 2 c Repliement de spectre

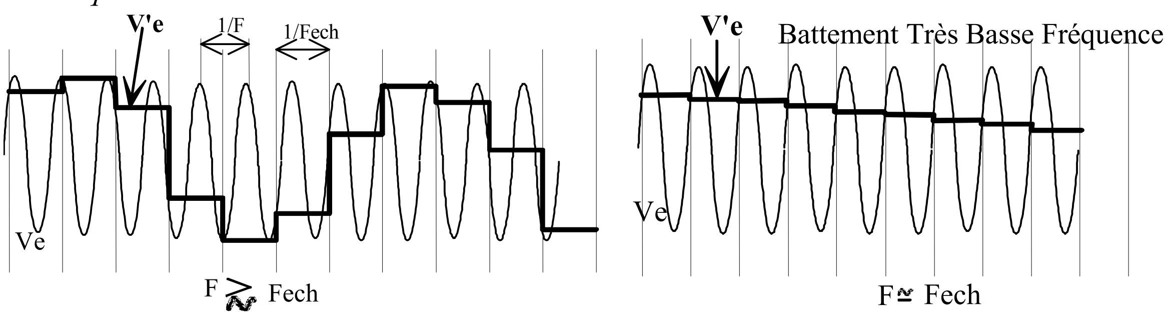
➢ ***Signal Ve sinusoïdal***

Pour une sinusoïde de fréquence F2 à l'entrée et si F2 > Fech /2 :



Le filtrage récupère la fréquence Fech-F2 et non F2 !!

Etude qualitative autour de Fech:



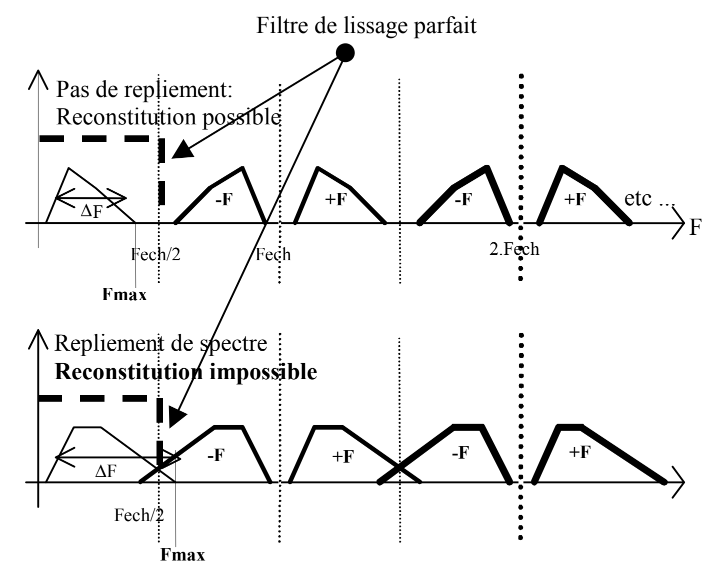
Pour F ≈ Fech, la fréquence de V'e est Fech - F ≈ 0 !!!. C'est un "battement" à très basse fréquence ou à fréquence nulle (si F = Fech), que l'on voit sur la figure ci-dessus.
Ce phénomène se retrouverait autour de 2.Fech de 3.Fech ... de k.Fech !

Application du repliement de spectre: Ce phénomène de battement en basse fréquence est mis à profit par les oscilloscopes à échantillonnage (permettant de visualiser des signaux de 50Mhz à plusieurs dizaines de GHz). On l'utilise maintenant également dans certaines techniques de démodulation AM FM en radiocommunication.
On peut citer aussi l'effet stroboscopique qui est en fait un échantillonnage par éclairs lumineux.

➢ ***Cas d'un signal de largeur de bande ΔF:***

Le **filtre** de **lissage** (**F2** du schéma général), limitant la bande à Fech/2, ne permet de *reconstituer le signal* analogique de départ que *dans le cas ou Fmax < Fech/2.*

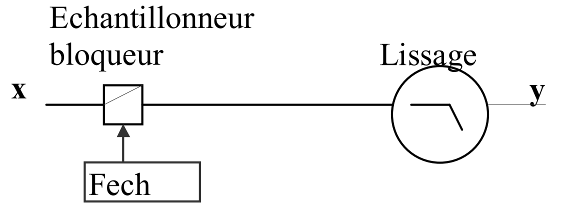
Le rôle du **filtre F1** optionnel (du schéma général) est de limiter la bande à Fech/2 ***avant échantillonnage***, afin d'éviter l'apparition de fréquences parasites, on le nomme filtre **"anti repliement"**. Il sert aussi, dans le cas d'un passe bande, à supprimer la composante continue qui doit souvent être nulle en traitement de signal.



Le signal de la figure du haut à un spectre avec Fmax < Fech/2 donc F1 n'est pas utile.
Le signal de la figure de bas à un spectre dépassant Fech/2, si on l'échantillonne sans filtrage initial, des battements parasites surviendront et ceux ci ne pourront plus être éliminés par la suite, reconstitution impossible !
Ce filtre anti repliement est ainsi indispensable dans de nombreuses applications, comme par exemple avant échantillonnage du signal audio en vue d'enregistrement sur compact disque.

I 2 d Fonction de transfert en fréquence de l'ensemble échantillonnage et lissage.

Soit la chaîne suivante, avec un signal d'entrée sinusoïdal $x = X.\cos(2\pi ft)$

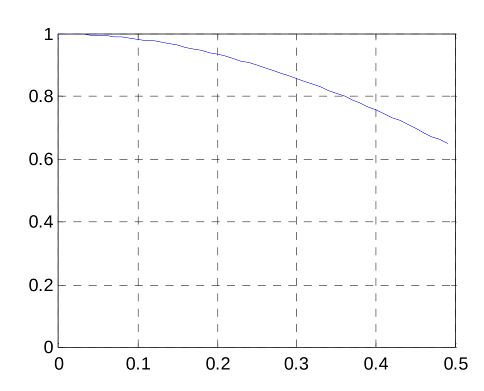


On pourrait montrer facilement (porte de largeur Te) que le bloqueur à lui seul possède une fonction de transfert qui atténue déjà légèrement le signal avant Fech /2 : son module vaudrait en effet :

$\frac{Sin(\pi x)}{\pi x}$ avec $x = f/fech$.

Ce défaut peut en fait si nécessaire se corriger numériquement. Il n'est pas forcément gênant.

D'autre part la fonction de transfert du filtre réelle du filtre de lissage atténue elle aussi légèrement avant Fech/2.



Une mesure de la simple chaîne acquisition → restitution peut donc s'effectuer, et on peut en tenir compte pour corriger les mesures sur un filtre numérique. Un filtre numérique ne pouvant pas aisément sur notre chaîne s'étudier seul.

I 3 Définitions diverses

I 3 a Rapport signal sur bruit d'un signal

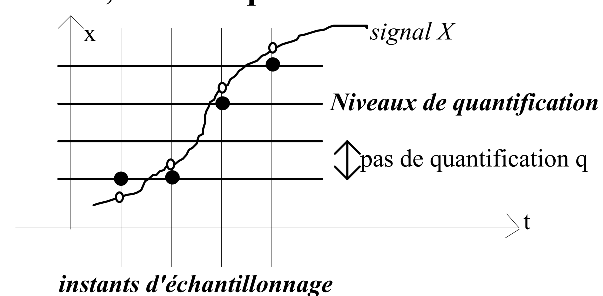
Soit un signal **X** constitué du signal utile **x** et d'un bruit **b**: **X = x + b**

En traitement du signal on parle peu d'erreur maximum ou de précision, mais de rapport signal à bruit exprimé en **dB**, qui est un rapport d'énergie (valeurs efficaces).

Il vaut $S/B = 10.\log\frac{x^2_{eff}}{b^2_{eff}}$ Et si le bruit est faible $S/B \approx 10.\log\frac{X^2_{eff}}{b^2_{eff}}$

Le bruit peut être du bruit seul, mais aussi des harmoniques provenant de distorsions. On parle alors de rapport S/(bruit+distorsion).

I 3 b Numérisation, bruit de quantification



● = échantillons Xk choisis au plus près par le CAN

o = valeurs exacte X du signal aux instants d'échantillonnag

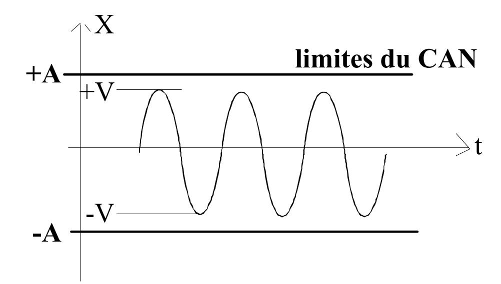
La différence y = Xk-X constitue une erreur **centrée**, comprise entre -q/2 et +q/2 provenant de la quantification.
On la nomme bruit de quantification.
On pourrait montrer que la valeur efficace de ce bruit s'exprime par **b^2eff = q^2/12**
Prenons maintenant une sinusoïde d'amplitude V, échantillonnée, puis convertie par un CAN de n bits, travaillant entre -A et +A.

- ***Raisonnement simple en Erreur Absolue Max sur Xk*** :

Il faut évidemment exploiter le plus possible le CAN (sans toutefois écrêter !), donc V voisin de A.

$$\text{Erreur absolue max} = \pm q/2 = \pm \frac{1}{2} LSB = \pm \frac{A}{2^n}$$

(LSB = poids du bit de faible poids du CAN).



- ***Raisonnement en Rapport Signal sur Bruit***

Signal : X^2eff = $V^2/2$

Pas de quantification : $q = A/2^{n-1}$ **Bruit :** y^2eff = $q^2/12 = \frac{A^2}{3.2^{2n}}$

Le rapport signal sur bruit vaut donc S/B = 10.log (X^2eff / y^2eff)
S/B = $10.\log(2^{2n}) + 10.\log(3/2) + 20.\log(V/A)$
Donc : **S/B = 6,02 n +1,76 - 20.log(A/V)** avec **A >= V**

On voit, ce qui était déjà évident, que S/B est maximum si on exploite toute l'excursion du CAN. Dans le cas ou **V = A** on a **S/B = 6,02 n +1,76**

I 3 c Traitements temps réel

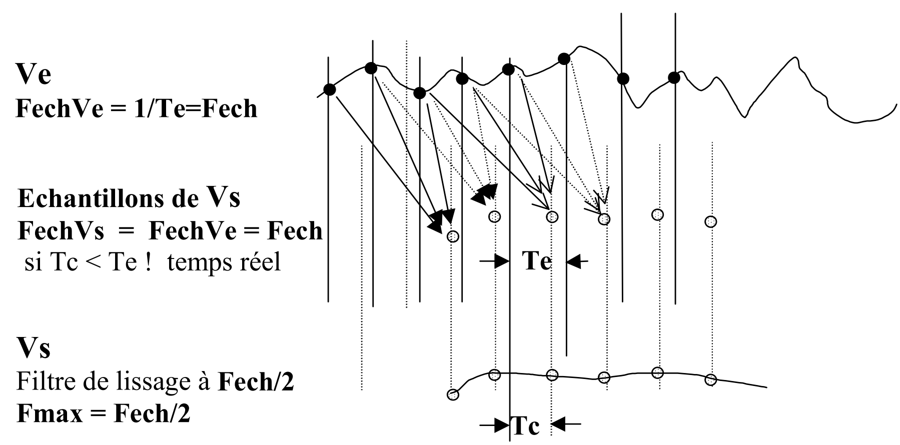
➢ **Traitement "en ligne" ou "au fil de l'eau ...": vrai filtrage numérique**

Soit Tech la période d'échantillonnage du signal X à traiter. Ce traitement consiste en trois phases: **acquisition** d'un échantillon **Xn**, **traitement** (à partir de Xn, et d'un certain nombre d'échantillons précédents pouvant provenir de différentes sources), **sortie** d'un échantillon **Yn** du signal de sortie Y.

On doit toujours conserver la même cadence d'échantillonnage pour le signal de sortie traité Y (**Fech entrée = Fech sortie),** l'échantillon Yn devra être fourni avant l'acquisition du nouvel échantillon Xn+1. Il faudra donc que

Tacquisition + Tcalcul + Tsortie < Tech.

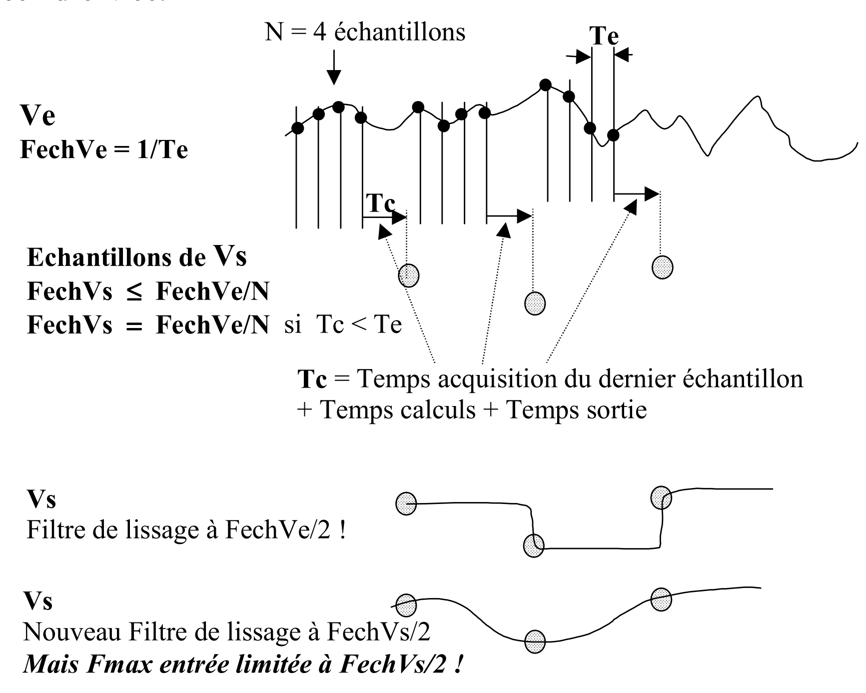
Filtre sur N = 3 échantillons



➢ **Traitement par bloc**

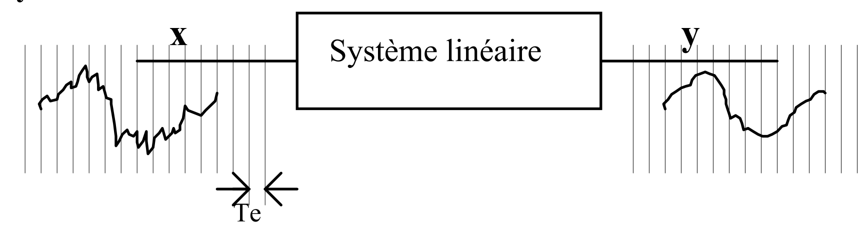
On fait l'acquisition de N valeurs du signal X. Le traitement s'effectue ensuite.

On travaille parfois ainsi, mais ce n'est pas du filtrage numérique ! Fech de sortie est inférieure à Fech d'entrée.



II Base du filtrage numérique, la convolution

II 1 a Système linéaire



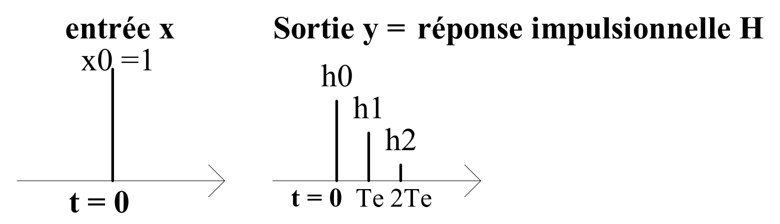
instants d'échantillonnage identiques pour tous les signaux: t = nTe = n/Fe

Linéarité: si x = x1+k.x2 alors y = y1 + k.y2
Invariant dans le temps: x(t-τ) donne y(t-τ)
Certains systèmes varient dans le temps, mais si cette variation est lente, on peut les classer dans cette catégorie.
On ne s'intéresse désormais qu'aux valeurs aux instants d'échantillonnage **nTe**.

II 1 b Réponse impulsionnelle (d'un système linéaire)

Si l'entrée **x** est une seule impulsion valant 1 pour t = 0, la réponse du système se nomme la <u>réponse impulsionnelle</u> **H** .
Elle peut être de **durée finie** ou **infinie :**
On distingue en effet deux types de filtres :
- **FIR** Finite Impulse Response
- **IIR** Infinite Impulse Response

Exemple simple:
Une réponse de trois échantillons (les suivants éventuellement négligeables)



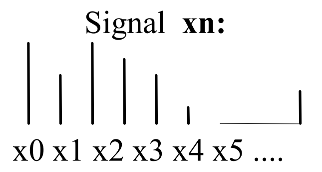
Ce filtre est évidemment « ***à réponse impulsionnelle fini***e » ou « **FIR** »

II 1 c Construction de la réponse yn à une suite d'échantillons xn. Convolution

Chaque échantillon xk, d'amplitude **xk**, donne une réponse égale à **xk.H**
Toutes ces réponses sont à sommer avec le décalage approprié.

On peut se représenter cela très facilement au moyen de l'exemple précédent qui est un filtre à réponse impulsionnelle finie de N = 3 échantillons:

Signal **xn:**



Il faut sommer: x0 × h0 h1 h2

+ x1 × h0 h1 h2

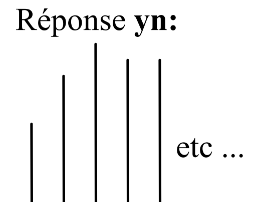
+ x2 × h0 h1 h2

+ x3 × h0 h1 h2

+ x4 × h0 h1 h2

etc

Réponse **yn:**



On a

y0 = x0.h0
y1 = x1.h0 + x0.h1
y2 = x2.h0 + x1.h1 + x0.h2
y3 = x3.h0 + x2.h1 + x1.h2
y4 = x4.h0 + x3.h1 + x2. h2
etc ...
$y_n = x_n.h0 + x_{n-1}.h1 + x_{n-2}.h2$ terme général

Soit **N la taille** du filtre. Pour un filtre à réponse impulsionnelle infinie, l'expression finale sera identique avec N tendant vers l'infini.
A partir de y_3 (indice = **taille N** de la **réponse impulsionnelle H**), on obtient y en sommant toujours N=3 termes.

On voit que la sortie se calcule en permanence à partir de x_n et des N-1 = 2 échantillons précédents, par l'expression:
Cette expression se nomme la **Convolution** de x avec h. On écrit parfois **y = x * h** (produit de convolution).

$$y_n = \sum_{k=0}^{N-1} x_{n-k} . h_k$$

Conclusion :

- A l'arrivée d'un signal x_n, un **régime transitoire** dure donc exactement **NTe** soit la **durée de la réponse impulsionnelle**.

A tout instant, la sortie est fonction de la nouvelle entrée x_n et de N-1 valeurs de l'entrée précédente. Donc prédictive exactement.

- Pour un **filtre IIR**, N est infini, le <u>régime transitoire est 'théoriquement' infini</u>, <u>mais heureusement pas en pratique !</u> c'est un peu comme une exponentielle en analogique qui bien que théoriquement infinie, ne varie pratiquement plus au bout de 4 à 5 constantes de temps !

III Base du filtrage numérique, réponse en fréquence, fonction de transfert

III 1 Etude générale avec x sinusoïdale, filtre de réponse impulsionnelle H

$x = A.\cos(2\pi ft)$

$x_n = A.\cos(2\pi nf/Fe)$ ou **$x_n = A.\cos(2\pi n.fTe)$**

On pose x = fréquence normalisée: **$x = f/Fe = fTe$** sans dimension.

$x_n = A\cos(2\pi nx)$

$0 \leq x \leq 0.5$

$0 \leq f \leq Fech/2$

Comme en analogique, pour calculer une fonction de transfert **G** qui apportera atténuation et déphasage en fonction de f, on introduit la notation complexe:

Soit $Xn = Ae^{2\pi inx}$ avec $xn = Reel(Xn)$

Alors :

$$G(x) = \frac{Yn}{Xn} = |G(x)|.e^{i.\varphi(x)}$$

G(x) est la **fonction de transfert complexe** du système

Le module du gain |G(x)| peut s'exprimer en dB: $G_{dB} = 20.\log(|G(x)|)$

Le déphasage est $\varphi(x)$

Pour une réponse impulsionnelle H, on obtient la sortie par la convolution : $Y_n = \sum_{k=0}^{N-1} X_{n-k}.h_k$

Or : $X_{n-k} = Xe^{2\pi inx}.e^{-2\pi ikx}$

Donc Xn-k signifie **donc déphasage** de **$-2\pi kx$** $= -2\pi kfTe$

Rappel : Si une sinusoïde $\sin 2\pi ft$ est retardée de τ, elle s'écrit : $\sin 2\pi f(t-\tau) = \sin(2\pi ft - 2\pi f\tau) = \sin(2\pi ft - \phi)$

Un **retard τ** provoque donc un **déphasage** de **$-2\pi f.\tau$**

Donc Xn-k signifie **donc un retard** de **k.Te**

Il vient

$$Y_n = \sum_{k=0}^{N-1} X_{n-k}.h_k = \sum_{k=0}^{N-1} Xe^{2\pi inx}e^{-2\pi ikx}.h_k = Xn\sum_{k=0}^{N-1} e^{-2\pi ikx}.h_k$$

D'ou

$$G(x) = \frac{Yn}{Xn} = \sum_{k=0}^{N-1} e^{-2\pi ikx}.h_k$$

On pose souvent (pour une étude plus générale dite transformée en z) : $Z = e^{2\pi ix}$

Et alors :

$$G(x) = \frac{Yn}{Xn} = \sum_{k=0}^{N-1} z^{-k}.h_k$$ avec $$z = e^{2\pi ix}$$

III 2 Gain pour f = 0 (donc pour le continu)

On fait f = 0, donc **x = 0** ou **z = 1**, il vient immédiatement :

$$G_0 = \sum_{k=0}^{N-1} h_k$$

Pour avoir donc un gain continu ou très basse fréquences, de **1**, il faut que la somme des coefficients soit égale à **1**

IV Partie pratique avec logiciel « Matlab »

Matlab est un outil très puissant utilisable dans de nombreux domaines de l'électronique, de l'automatisme, etc ….
La syntaxe ne ressemble hélas pas du tout au C, on **travaille** toujours **sur des Matrices** !
Pour ne pas être obligé d'apprendre ici un nouveau langage, mais pour en entrevoir toutefois quelques unes de ses possibilités, les programmes à écrire vous sont fournis et commentés, il ne vous restera plus qu'à les écrire et à modifier certaines valeurs.

Mode d'emploi très succinct de Matlab
On écrira des nouveaux programmes Matlab, les sauver par exemple en essai1.m etc
Les fenêtres de base de Matlab :

Command Window, au signe >
- Pour exécuter taper nom **nom du programme.m**
- Pour voir les valeurs d'une variable taper son nom.
- Flèche ↑ retour à la commande écrite précédente

Editeur/Debugger Window : Pour l'édition du programme
Fenêtres figures chronogrammes en exécution.

IV 1 Etude d'une simple convolution

On désire effectuer la convolution entre :
Les échantillons xn 0 1 2 3 4 5 5 5 5 5 0 0 0 0
La réponse impulsionnelle : **H** **0,6 0,3 0,1** (trois échantillons h0 h1 h2)

L'entrée s'établit à t = 0 en **rampe** (de pente 1), puis est **constante** (**à 5**) et enfin est **constante** (**à 0**). Le signal d'entrée se modifie donc 3 fois, on devra donc observer **3 régimes transitoires** et **trois régimes permanents**

IV 1 a Etude théorique à la main

1) Compléter le tableau suivant (détacher celui fourni en dernière page …), en s'inspirant de **l'explication physique** toute simple de la convolution, somme de réponse impulsionnelle, donc **ligne par ligne**, on ajoutera ensuite verticalement.

Xn ↓	Te	2Te	3Te	4Te	5Te	6Te	7Te	8Te	9Te	10Te	11Te	12Te
0	0	0	0									
1		0,6	0,3	0,1								
2			1,2	0,6	0,2							
3												
4												
5												
5												
5												
5												
5												
0												
0												
0												
0												
Yn→	0	0,6	1,5									

2) Tracer sur un même chronogramme (également sur la feuille en dernière page) les valeurs des Xn et des Yn (marquer bien les différents points et les relier par une ligne pointillée pour mieux observer l'allure générale du signal.

Bien indiquer sur ce chronogramme les **régimes permanents** et les **régimes transitoires**.
Remarque : les transitoires durent théoriquement NTe = 3Te mêmes si certains paraissent plus courts…

3) Que vaut la valeur maximale de Yn, est-ce normal et pourquoi ?

IV 1 b Vérification sur Matlab

1) Ecrire le programme suivant (Vous n'êtes pas obligé d'écrire les commentaires…)

```
x = [ 0 1 2 3 4 5 5 5 5 5 0 0 0 0 ];     % Tableau des xn
h = [0.6  0.3  0.1 ];                     %Réponse impulsionnelle de trois coefficients
y = CONV(x,h);                            % Convolution de x et de h

plot(x,'k*-');                            % Tracé des xn, en noir, étoiles, reliés par trait
grid on;                                  % quadrillage
hold on;                                  % pour tracer une seconde courbe sur la même figure
plot(y,'r*:');                            % Tracé des yn, en rouge, étoiles, reliés par pointillés
hold off;
```

2) Exécuter le programme. Est ce Ok rapport à construction manuelle précédente ?
3) Taper y et vérifier les valeurs obtenues manuellement.
4) Indiquer sur la figure obtenue les régimes permanents, et vérifier bien la durée des transitoires
5) Si M et N sont respectivement la taille du tableau des Xn, et la taille de la réponse impulsionnelle. Quelle est la taille de la convolution Y ?

IV 2 Observation d'un régime sinusoïdal

Soit à t = 0 un signal x sinusoïdal de 370Hz, échantillonné à 10000 HZ, il est envoyé vers un filtre de réponse impulsionnelle h formé par 7 coefficient égaux à 1/7

IV 2 a Partie théorique

Quel est le gain en continu ?
Quelle est la durée du régime transitoire (en **nombre de Te** puis **en ms** ?)

IV 2 b Observation sous matlab

1) Ecrire sous Matlab le programme suivant :

On remarquera le tableau génère t de 50 points espacés de Te. (Commentaires inutiles ….)

```
fe=10000;               % Fréquence d'échantillonnage 10000 Hz
t = 0:1/fe:50/fe;       % 50 points espacés de Te = 1/fe
f = 370;                % Fréquence du Signal,   370 Hz
x = cos(2*pi*f*t);      % Signal sinusoïdal
h=1/7*[ 1 1 1 1 1 1 1 ];    % Réponse impulsionnelle :   7 coefficients égaux à 1/7

y =CONV(x,h);           % convolution
plot(x,'k*-');          % Tracé des xn
grid on; hold on;
plot(y,'r*:');          % Tracé des yn  sur même figure
hold off;
```

2) Observer les courbes obtenues.

Bien marquer sur les chronogrammes les **deux régimes transitoires** et le **régime permanent**. Pour celui ci, on observera simplement gain et déphasage, une étude plus précise est faite dans un TP ultérieur.

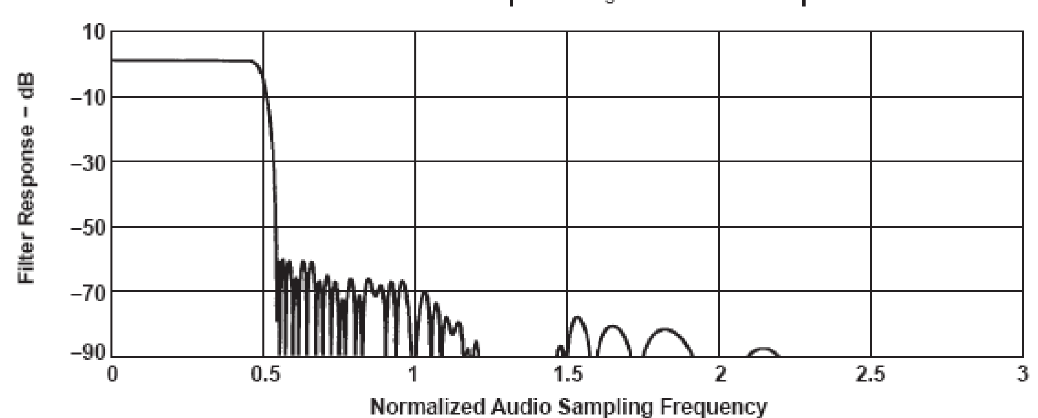
3) Vérifier la durée des régimes transitoires, est ce normal ?

V Découverte d’une chaîne de traitement de signal à DSP (Digital Signal Processeur) : starter Kit TMS320C6713 DSK

V 1 Description succincte de la maquette

- Elle contient toute la chaîne décrite au début :
  - Entrée signal analogique **deux voies.**
  - Filtre **anti-repliement F1 ici toujours présent** (possibilité de le supprimer sur d’autres cartes).

PARAMETER	TEST CONDITIONS	MIN TYP MAX	UNIT
ADC Filter Characteristics (TI DSP 250 f_S Mode Operation)			
Passband	±0.05 dB	0.416 f_S	Hz
Stopband	−6 dB	0.5 f_S	Hz
Passband ripple		±0.05	dB
Stopband attenuation	f > 0.584 f_S	−60	dB

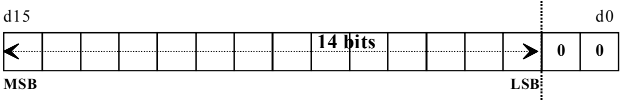


  - Echantillonnage à **Fech = 48 kHz (pour ces TP).**
  - Conversion par **CAN**, **deux voies** simultanées.
  - Traitement par le DSP texas : **TMS320C67**, **Fck = 225 MHz**
  - **CNA deux voies** simultanées**.**
  - Filtre **de lissage F2,** caractéristiques voisines de F1, sortie analogique **deux voies.**
  - **Certaines cartes** peuvent avoir un **passe haut supplémentaire** en entrée et en sortie commençant à couper en dessous de 50 Hz (simple capa de liaison).

- **Interfaces Analogiques :**

Fréquence d’échantillonnage programmable, et position du filtre de lissage automatiquement à Fech/2.

Convertisseurs **n bits (n <=16).** Données **cadrées à gauche**, donc de dynamique presque identique quel que soit ce nombre de bits. Exemple de cadrage gauche pour 14 bits:

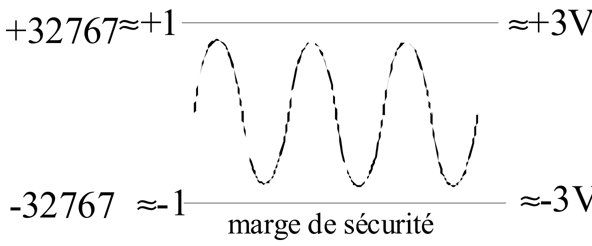


Tensions analogiques :

Entrée **x(t)** et sortie **y(t)**: **max**imum **1,3 volts crête**

Echantillons **xn** et **yn** sur **16 bits signés (mode complément à 2)**, donc en n’utilisant pas –32768, dynamique de :

- De **0 à ±32767** en raisonnement entier
- Ou de **0 à presque ±1** en raisonnement 15 bits fractionnaires : Q15(16)



-,--- ---- ---- ----

→ Eviter de dépasser en entrée et en sortie les valeurs maximales, sinon des distorsions plus ou moins brutales non étudiées ici interviendraient (repliement brutal d'amplitude, écrêtage).

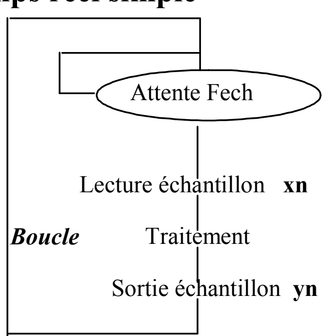
➢ ***Le processeur est un DSP virgule flottante :***
Ce composant est très rapide même en travaillant sur des flottants, son unité arithmétique interne étant conçue pour travailler directement sur une mantisse et un exposant.
Un DSP virgule fixe pourrait certes travailler aussi sur des flottants, mais au moyen de librairies de calcul et donc demanderait 10 à 100 fois plus de temps par opérations !

→ Il n'y a pratiquement pas de problèmes de débordement en cours de calcul sur un DSP virgule flottante, mais quelques précautions subsistent tout de même :
→ Si le DSP doit fournir des valeurs à un CNA câblé comme le CAN d'entrée en cadrage gauche, on ne doit pas envoyer d'échantillons plus grands que –32767 et + 32767, sinon la conversion en entier est tronquée provoquant ainsi des débordements et des signaux d'apparence n'importe quoi !
→ ***Conclusion :***
On connaît la dynamique du signal d'entrée x.
Il faut considérer seulement la **dynamique pratique du signal y fourni au CNA,** pour cela :
Prévoir le gain maximum de la chaîne en fonction de la fréquence.
Prévoir aussi les dépassements lors d'un transitoire.
(En effet un filtre peut très bien avoir un module de gain toujours inférieur à 1 présenter un transitoire un peu oscillant et dépassant ce gain de 1).
Il faut parfois ne sortir au CNA que la moitié, le quart (ou même moins) du signal y calculé.

V 2 Organigramme général d'un traitement en temps réel simple

Pour assurer le temps réel, toute la boucle doit avoir une durée inférieure à Tech=1/Fech !

Tboucle < Tech



V 3 Variante pour cette maquette et pour logiciel fourni

En réalité, l'ensemble fonctionne avec un logiciel plus compliqué de travail par « bloc » non étudié ici, avec un méthode dite de « double buffer » en deux phases: **acquisition et sortie** (**IO**) d'échantillon automatique sans passer par le DSP (en DMA : **D**irect **M**emory **A**ccess) d'un coté (buffers Ping), et en même temps d'un autre coté traitement par le DSP (buffers Pong), et inversement.

Phase 1 : IO sur PING | **Phase 2 : IO sur PONG**

Phase 1 : Acquisition Restitution ; xk → In Ping (0 … N-1) ; Out Ping → yk ; In Pong (0 … N-1) → Traitement → Out Pong

Phase 2 : In Ping (0 … N-1) → Traitement → Out Ping ; xk → In Pong (0 … N-1) ; Out Pong → yk ; Acquisition Restitution

On n'exploitera pas vraiment les possibilités du double buffer, mais on travaillera dans un mode correspondant à l'organigramme suivant :

Soit BLOCKSIZE la taille de bloc.
La durée de traitement de **tous** les échantillons du bloc doit être inférieure à :
BLOCKSIZE*Tech

Et ce qui revient au même :
Tboucle < Tech

→ ***Mode de travail simplifié***

Interruption fin d'acquisition **d'un bloc**

Pour tous les échantillons de **l'autre bloc**

Traitement_échantillon

Boucle

→ **Les IO d'échantillons** s'effectuant en **tache de fond** par interruptions, on ne s'en occupe absolument pas.

V 4 Projet et programme de démonstration fourni

Dans le sous répertoire : C:\CCStudio**MyProjects\dsp_c67 ini GEII**
Ouvrir le projet : **dsk_c67.pjt**
Aller dans **sources**, il vous faudra les fichiers C principaux (et seulement ceux ci) :

dsk_c67.c	le noyau de l'application, ne rien modifier !
aic23.c	fonctions de l'interface analogique, ne rien modifier
demo.c	**Le fichier C de travail** (si ce fichier n'est pas présent

dans le projet, ou si il y en a un autre, l'ajouter et ne garder comme fichiers.C dans le projet que ces trois fichiers. Ne rien modifier d'autre évidemment !

Fichier **demo.c** : Le traitement est ici tout simple. On ne fait qu'acquérir un signal à la fréquence Fech, le numériser, le multiplier par **1,5** et le reconstituer.

```
//                              PROJET toujours dsk_c6713.pjt
// kit de développement TEXAS TMS 320C6713 DSK
// I[0] O[0]: voie gauche ---> "tip" ;  fiche RCA blanche ou noire
// I[1] O[1]: voie DROITE ---> "ring" ; fiche RCA ROUGE
// Dynamique des échantillons:
//          +- 1,3 Volt crète
//          -32767 à +32767    en Q0(16)         (sur des int de 32 bits)
//          presque (-1 à +1)  en Q15(16)  ( Q31(32) en fait )
//Si on veut travailler sur des flottants de -1 à +1,
//                    On divisera par 32768.0  en entrée
//                    On re multipliera par 32768.0 avant de sortir
// //@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
#include <log.h>
#include <math.h>
extern far LOG_Obj trace;   // non utilisée ici
void pulse_gpio2(void);   // pour mesurer des durées en temps réel

// Initialisations avant la boucle principale de travail
Init_traitement()        // rien ici
{
}
// Traitement des échantillons comme une simple acquisition un par un (2 voies)
//  Produit par 1.5 en flottant
```

```
void traitement_echantillons(int *I, int *O)        // [0] voie G      [1] voie D (rouge)
{
O[1] = 1.5 *(float) I[1];   // voie droite, indice 1
}
void periodic_log(void)                // non utilisé ici
{ }
```

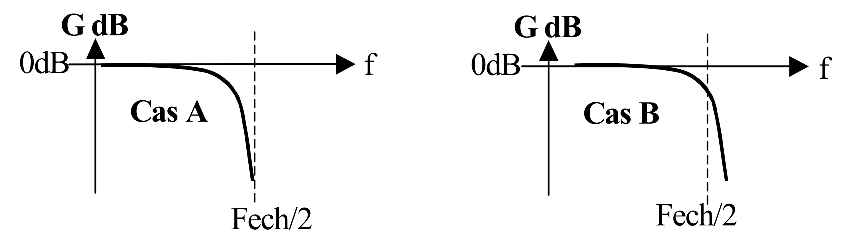
→ On remarque en fait deux fonctions que l'on peut modifier :

Init_traitement() Servira pour initialiser des valeurs avant le traitement principal.
void traitement_echantillons(int *X, int *Y) La fonction de traitement correspondant à votre application

→ **Ne cherchez pas de programme main() !** il se trouve dans un autre fichier (dsk_c76.c).

V 5 Positions des filtres F1 et F2 et questions théoriques

→ Les filtres F1 et F2 ont une raideur et une position fixe (par rapport à Fech/2) déterminées par le constructeur.
Mais il y aurait en fait deux choix possibles pour le positionnement de ces filtres:



Cas A : Forte atténuation à Fech/2. On privilégie une bonne protection contre le repliement de spectre (filtre F1), et un bon lissage (filtre F2). Mais en contrepartie la bande passante est réduite.
Cas B : Début d'atténuation à Fech/2. On privilégie la bande passante, en tolérant des fréquences parasites quand on s'approche de Fech/2, et un moins bon lissage.

→ Remarque, en **Haute Fidélité** (HIFI) on est souvent dans le **cas B**, car en fait les fréquences parasites présentes sont inaudibles, et la bande passante est ainsi améliorée pour une même Fech.

D'après les caractéristiques des filtres fournies précédemment :

1) Etes-vous dans le cas A ou dans le cas B, et que privilégie-t-on (meilleur lissage ou meilleure bande passante) ?
2) Donner la fréquence **Fmax** où l'atténuation reste encore négligeable.
3) Donner la fréquence $\mathbf{F_{-6db}}$ ou l'affaiblissement est de 6dB. Quelle est la valeur du **module** de la fonction de transfert pour cet affaiblissement (cette valeur servira dans la partie pratique pour mesurer en ce point).
4) Donner la fréquence $\mathbf{F_{-60db}}$ au delà de laquelle l'affaiblissement est > 60dB

V 6 Travail pratique

Eviter toujours de déformer les signaux en ne mettant pas trop d'amplitude de signal !
De plus, du fait du produit par 1,5 si vos échantillons de sortie dépassent la valeur max sur 16 bits (±32767), vous verrez très vite une forte distorsion (débordement) si vous dépassez !

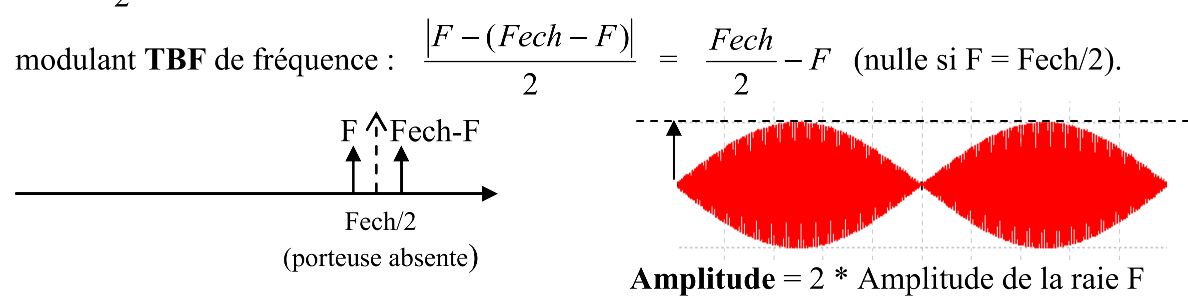
V 6 a Relevé du module de gain de l'ensemble : échantillonneur et filtre de lissage, pour f <= Fech/2 (<=24 kHz)

1) Remarques :

→ Lorsque F augmente et du fait de l'échantillonnage avec maintient (voir cours théorique), on observera déjà une petite atténuation, et à un moment donnée celle ci augmentera brusquement (intervention des filtres F1 et F2).

→ Lorsque F se rapproche de Fech/2 et du fait du repliement de spectre, F et Fech-F deviennent proches. Deux cas peuvent se présenter :

Si ces fréquences ne sont pas encore trop atténuées (filtres F1 et F2 du cas B), on peut visualiser alors nettement un mélange de ces deux fréquences. Ces deux fréquences étant alors voisines, le signal observé ressemble à un signal modulé AM : porteuse centrale $\frac{F+(Fech-F)}{2} = Fech/2$ non présente, et deux bandes latérales F et Fech-F, donc signal modulant **TBF** de fréquence : $\frac{|F-(Fech-F)|}{2} = \frac{Fech}{2} - F$ (nulle si F = Fech/2).



Amplitude = 2 * Amplitude de la raie F

Si ces fréquences sont très atténuées (filtres F1 et F2 du cas A), on retrouve ce même signal très faible et mélangé à du bruit, on ne peut rien en tirer ni mesurer vraiment une atténuation, on peut dire éventuellement que celle ci est supérieure à une certaine valeur.



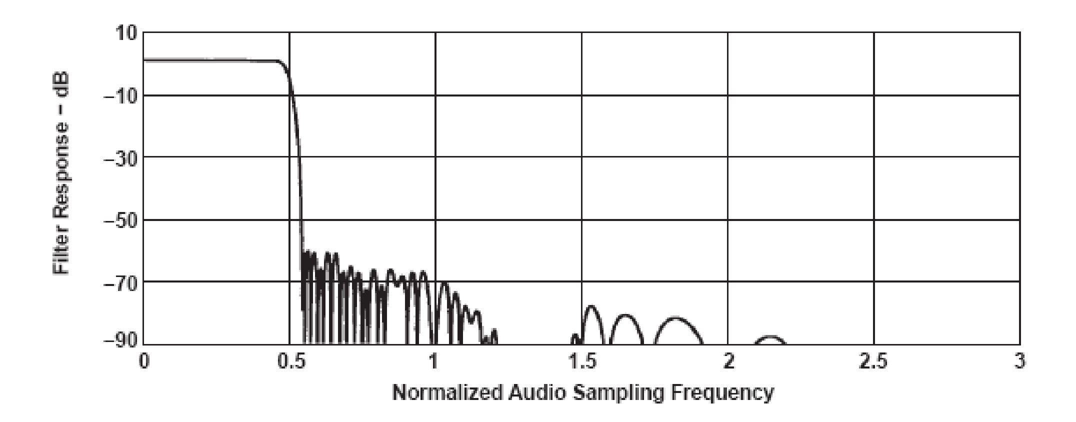
2) Mesures

On fera les mesures avec un générateur sinusoïdal réglé sur **environ 500 mVolt Crète**. Et on visualisera sur l'oscilloscope x(t) et y(t). En faisant varier la fréquence de 0 à Fech/2, observer et tracer (en <u>échelles linaires</u>) l'<u>allure</u> de la courbe de réponse en fréquence.

On connaît ici à priori la fonction de transfert que l'on doit vérifier (voir caractéristiques des filtres F1 et F2 fournies plus haut). Donc **ne pas mesurer inutilement des centaines de points n'importe où** ! **quelques mesures** bien placées sont suffisantes: une ou deux **en basse fréquence**, une **au milieu** de la bande passante, puis aux fréquences théoriques trouvées précédemment sans dépasser pour l'instant Fech/2 = 24 kHz.

3) Comparaison avec la théorie

On redonne ici les caractéristiques théoriques de F1 et F2. Vous avez déjà calculées dans la partie théorique les fréquences importantes dans votre cas.



- On peut déjà observer une différence importante, laquelle, expliquez d'où cela peut-t-il provenir.
- Comparez maintenant toutes vos mesures aux valeurs que vous avez déduites de la doc constructeur. Conclure.

4) « Shannon théorique » : rappeler la condition de Shannon pour pouvoir reconstituer et retrouver le signal de départ. Valeur numérique ?

5) « Shannon pratique » : estimer en pratique sur cette maquette la bande de fréquence (en Hz et de 0 à ?) sur laquelle vous aller pourrez réellement travailler, c'est à dire où l'ensemble « échantillonnage et restitution » peut être considérée comme sans influence (peut être à l'atténuation près due au principe même de l'échantillonneur bloqueur, que l'on pourrait d'ailleurs corriger assez aisément par un petit filtre si nécessaire).

V 6 b Observation pour f > Fech/2 (f > 24 kHz)

1) Continuer la courbe précédente en relevant le module du gain pour 25kHz, 27kHz et 30 kHz par exemple.
2) Conclure en l'expliquant sur la présence ou non du filtre F1 anti repliement.

V 6 c Mesure de la fréquence d'échantillonnage exacte, Fech

Il y a deux méthodes plus ou moins applicables selon les cas :

Mesure à F ~ Fech : On observe le battement de fréquence nulle (battement zéro) pour F = Fech. Pour cela, il ne faut évidemment ne pas avoir à l'entrée de filtre anti-repliement. On peut mesurer ainsi Fech à 1 Hz près si on veut !

Mesure à F ~ Fech/2 : Si l'atténuation n'est pas trop importante à Fech/2, on peut observer aisément le signal de sortie qui est équivalent à une modulation d'amplitude à porteuse supprimée, de fréquence de modulation très faible. On mesure ainsi Fech en trouvant le « battement zéro », à 1 Hz près si on veut !

1) D'après vos résultats précédents, sur votre système, quelle méthode allez vous pouvoir utiliser (en l'expliquant) ?
2) Effectuer cette mesure à quelques Hz près.

V 6 d Meilleure observation du repliement de spectre, pour f > Fech/2

Comme il **y a toujours sur votre maquette** à l'entrée **le filtre F1** « anti-repliement » qui limite la bande, on ne peut évidemment pas sur cette maquette et ce programme observer vraiment et complètement le repliement de spectre, pour F > Fech/2. Mais on peut ruser !

1) Enlevez demo.c de votre projet et le remplacer par **demo_sous_ech.c**

La fonction de traitement est ainsi remplacée par :

```
void traitement_echantillons(int *I, int *O)
{
static char k=0;   static float ech=0;   // pour conserver un échantilon lu
// sous échantillonnage rapport 4:   Fech entrée = 48/4 = 12 kHz
if(k++>=3) { ech = (float)I[1]; k = 0;}
O[1] = fir(ech);   // voie droite, sortie lissée  et lissage à 6 kHz
O[0] = ech;                  // sortie non lissée
}
```

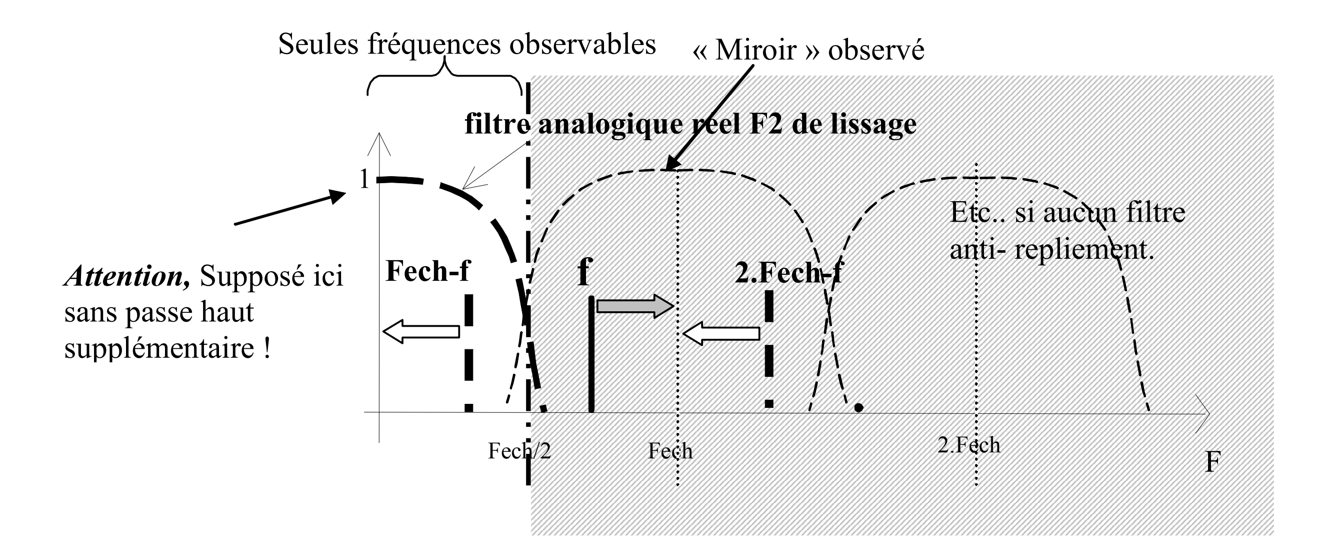
Fech est toujours de 48kHz, donc les filtres d'anti-repliement et de lissage interviennent toujours à 24kHz.
On a sous échantillonné dans un rapport 4, et on a ajouté sur la voie de droite un filtre de lissage numérique (FIR) à 6kHz (avec forte atténuation à 6kHz).

On a donc :	Nouvelle	**Fech = 12 kHz**
	Nouvelle	**Fech/2 = 6kHz**
	Nouveau **filtre de lissage en sortie, (F2) à 6kHz (forte atténuation à 6kHz)**	
Le Filtre anti repliement (**F1)** à l'entrée interviendra seulement à 2*Fech = 24 kHz		

2) En faisant varier la fréquence f du signal d'entrée de presque **0 à presque 2*Fech (ici 24KHz) tracer rapidement** (en ne mesurer que quelques points utiles !!!) le module de la fonction de transfert.

Pour F augmentant au-delà de Fech/2, on doit observer le « miroir » de la fonction de transfert précédente, ceci s'explique en fait par le « repliement de spectre » à l'entrée. La composante Fech – F du signal échantillonné xn évolue de Fech/2 à 0 (se déplace vers la gauche) et se retrouve ainsi dans le gabarit du filtre de lissage précédent. Ce n'est plus une vrai fonction de transfert, mais on peut parler de « pseudo fonction de transfert » et mesurer tout de même un module même si les fréquences à l'entrée et à la sortie ne sont plus les mêmes !

Au-delà de 24 kHz, on pourrait encore observer des miroirs, mais on ne verra ici plus rien le filtre anti repliement de départ (F1) étant toujours présent à 24kHz !!!



3) Pour f ≈ Fech, Fech-f ≈ 0, on doit donc **observer un battement très basse fréquence** de grande amplitude, par cette méthode, **mesurer** de nouveau la **fréquence d'échantillonnage à quelques Hz près**, est ce OK ?

4) Conclure sur la **nécessité** dans de nombreuses applications d'un filtre anti–repliement. On en décrira brièvement ses caractéristiques (pour être un bon filtre) en le comparant à celui du filtre de lissage. Bien comprendre que sans ce filtre, des fréquences assez élevées (du bruit par exemple) pourraient se retrouver en basse fréquence et distordre notablement notre signal utile.

Quelques applications utilisent néanmoins le « sous échantillonnage » sans filtre anti-repliement, afin de créer volontairement des changements de fréquences, par exemple pour des récepteurs de radio tout numérique.

Noms :

Xn ↓	Te	2Te	3Te	4Te	5Te	6Te	7Te	8Te	9Te	10Te	11Te	12Te
0	0	0	0									
1		0,6	0,3	0,1								
2			1,2	0,6	0,2							
3												
4												
5												
5												
5												
5												
5												
0												
0												
0												
0												
Yn→	0	0,6	1,5									

